International Journal of Approximate Reasoning 51 (2010) 869–882

Contents lists available at ScienceDirect

International Journal of Approximate Reasoning

journal homepage: www.elsevier.com/locate/ijar

Decision making with imprecise parameters

Asli Celikyilmaz [a,*], I. Burhan Turksen [b,c]

[a] Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA, USA
[b] Department of Mechanical and Industrial Engineering, University of Toronto, Toronto, Ont., Canada
[c] Department of Industrial Engineering, TOBB-Economy and Technology University, Sogutozu, Ankara, Turkey

ARTICLE INFO

Article history:
Available online 3 July 2010

Keywords:
Interval-valued membership functions and imprecise functions
Cased-based type reduction

ABSTRACT

We analyze the impact of imprecise parameters on performance of an uncertainty-modeling tool presented in this paper. In particular, we present a reliable and efficient uncertainty-modeling tool, which enables dynamic capturing of interval-valued clusters representations sets and functions using well-known pattern recognition and machine learning algorithms. We mainly deal with imprecise learning parameters in identifying uncertainty intervals of membership value distributions and imprecise functions. In the experiments, we use the proposed system as a decision support tool for a production line process. Simulation results indicate that in comparison to benchmark methods such as well-known type-1 and type-2 system modeling tools, and statistical machine-learning algorithms, proposed interval-valued imprecise system modeling tool is more robust with less error.

1. Introduction

Fuzzy systems are useful tools that can deal with complex, ill defined and uncertain environment parameters, where conventional mathematical models fail to give satisfactory results. In many research papers and books such as in [1–3] it has been shown that type-1 fuzzy models can have limitations in identifying uncertainties, since membership functions to characterize type-1 fuzzy sets are crisp, rather than fuzzy values. An extension of type-1 fuzzy sets, also known as higher order fuzzy sets, e.g., type-2 fuzzy sets, can be characterized with type-2 membership functions that are themselves fuzzy. Type-2 fuzzy sets are introduced by Zadeh [4] as an extension of type-1 fuzzy sets. It has been demonstrated in many different research papers [3,5,6] that with the implementation of higher order fuzzy sets to build type-2 fuzzy systems or interval-valued fuzzy systems, the performance of predicted model can be improved to a certain degree compared to type-1 fuzzy systems, e.g., [1,7–9]. Hence, in this paper we introduce and investigate a new interval-valued fuzzy system modeling strategy, in short interval-valued fuzzy functions (IVFF) approach, as an extension of our earlier type-1 fuzzy functions (T1FF) methodology [7,8,10].

In modeling real systems, implementing type-2 fuzzy sets instead of type-1 fuzzy sets can be useful when it is difficult to determine the exact and precise membership values of data points. Nevertheless, computations with general type-2 fuzzy sets are rather complex compared to type-1 fuzzy sets. For such reasons in the literature, interval-valued fuzzy inference systems, e.g., [1,6,11], which implement interval-valued type-2 fuzzy sets, have commonly been used instead of type-2 fuzzy sets to reduce the computation complexity to a certain degree. In such systems, *footprint of uncertainty* (FOU) is the general term used for interval-valued membership functions, to define the uncertainty interval of membership values. The FOU shows the uncertainty region of type-1 membership functions [14] by forming an interval that is bounded with upper

* Corresponding author. Tel.: +1 510 229 8269.
E-mail addresses: asli@berkeley.edu, asli@eecs.berkeley.edu (A. Celikyilmaz).

0888-613X/$ - see front matter
doi:10.1016/j.ijar.2010.06.002

and lower membership functions, which are usually identified by experts. Generally, each of type-1 membership function, which are embedded within the boundaries of FOU of interval type-2 fuzzy sets, is used to construct embedded type-1 fuzzy models. For such systems, a type-reduction method is implemented which follows a defuzzification algorithm to obtain crisp outputs where necessary.

In type-2 fuzzy inference systems (FISs), more usual than not, the shape or parameters of type-2 membership functions, boundaries of FOU, or structure of the rules, are identified by domain experts [1,14,23,24]. Manually designing and tuning the parameters of membership functions of type-2 fuzzy systems may result in false assumptions and probably affect predicted model's performance. Usually, pre-defined shapes e.g., Gaussian or triangular fuzzy sets, are used to define fuzzy sets. Such type-2 FISs implement mainly Takagi–Sugeno or Mamdani type fuzzy systems.

Interval-valued fuzzy inference system (IVFS), to be presented at a later point of this paper, uses a fuzzy clustering algorithm to identify overlapping regions that may exist in the dataset and assigns membership values for each data point in the dataset to each cluster. For different values of initial parameters of fuzzy clustering methods one may obtain different membership values. Hence, we construct embedded (discrete) interval membership values by iterating learning parameters of fuzzy clustering method. For each iteration we estimate a set of fuzzy functions, one for each cluster to capture the local input–output dependencies. Thus proposed IVFS dynamically identifies embedded fuzzy functions for each cluster using interval-valued fuzzy sets obtained from a fuzzy clustering algorithm of imprecise parameters, particularly using a hybrid clustering method improved fuzzy clustering (IFC) [8]. Different from the hybrid clustering methods [13] of literature, IFC is designed to shape membership values so that they can help to shape local functions to define input–output dependencies of each cluster. The aim is to find the optimum functions to minimize the error. The fuzzy modeling approach being presented in this paper is feasible in the sense that it implements a non-complex inference tool with a practical type-reduction method based on case-base reasoning.

The rest of the paper is organized as follows: In Section 2, we briefly review type-1 fuzzy functions in comparison to well-known type-1 fuzzy inference systems. In Section 3, we introduce the new interval-valued fuzzy function representation and inference modules. In Section 4, we present the results of experiments conducted on application of proposed and benchmark methods on a desulphurization process of steel production. Finally, we draw conclusions in Section 5.

2. Fuzzy inference system based on type-1 fuzzy functions

Before we explain the interval-valued fuzzy function methodology of this paper, in what follows, we briefly review foundations of type-1 fuzzy functions approach in comparison to well-known fuzzy inference systems. A detailed introduction and explanation of underlying theories of type-1 fuzzy functions can be found in [8,10].

2.1. General fuzzy inference systems

Traditional fuzzy representation and inference systems, i.e., either type-1 or type-2 [1,9,12,14] have various challenges that are in need for review for this paper. Among some of these challenges are;

- Identification of types of antecedent and consequent membership functions, and their varying parameters.
- Identification of the most suitable combination operators (t-norm, t-conorm, etc.), conjunction operators while aggregating antecedents, and consequents of each rule.
- Identification of the type of implication operator to capture uncertainty associated with linguistic "AND", "OR", "IMP" for representation of rules, and reasoning with them.
- Identification of the type of the defuzzification method.

These issues have been investigated in many research papers to optimize the fuzzy operations. Many different methods are proposed to optimize the parameters of such fuzzy systems and reduce expert intervention in building hybrid fuzzy systems by using other soft-computing methods such as genetic algorithms or neural networks, e.g., [15,16,18,25–28] as a parameter or structure optimization tool. Some researchers have approached latter issues differently and tried to reduce computation complexity and prevent information loss in a different way as introduced in [17], which was later utilized in several papers such as [3,7,8,10]. Such fuzzy systems are constructed under the assumption that antecedent fuzzy sets are dependent on each other, i.e., they are interactive; hence they characterize multi-dimensional membership functions to represent entire antecedent part of any rule. An extension of such method using Takagi–Sugeno [9] systems can be defined as follows:

$$R_i: \ IF\ x \in X\ is\ A_i\ THEN\ y_i = a_i x^T + b_i \tag{1}$$

In (1) fuzzy set A_i is characterized by a type-1 membership function $\mu_i(x) \rightarrow [0,1]$, which represents entire antecedent part of rule i and $x \in X$ is a multi-input vector. Since only one antecedent fuzzy set is defined for each rule, fuzzification would be simpler, aggregation of antecedent step is eliminated, and there is no independence assumption of input variables, which may affect the prediction performance of the overall fuzzy system model. In addition, we would eliminate the a possible performance decrease due to information loss that may be encountered when mapping the multi-dimensional membership functions onto each individual input dimension. Fuzzy system models based on fuzzy functions [7,8,10] also adopt interactivity of input fuzzy sets as explained in the next subsection.

2.2. Type-1 fuzzy functions

Type-1 fuzzy functions (T1FF) [7,8,10] approach is an alternate representation and reasoning tool to standard fuzzy inference systems. It can be considered as an extension of Takagi–Sugeno and Sugeno–Yasukawa fuzzy inference systems, except that such systems implement multi-dimensional antecedent fuzzy sets. Structure identification of T1FF is based on a fuzzy clustering algorithm, e.g., fuzzy c-means (FCM) [19] algorithm or improved fuzzy clustering method [8] to find possible hidden structures of a given dataset and characterize input membership distributions to represent multi-dimensional membership functions of input domain. These methods do not require most of the aforementioned fuzzy operations of traditional fuzzy inference systems as mentioned in the previous subsection. In somewhat simplified view, type-1 fuzzy function architecture is demonstrated in Fig. 1.

As shown in Fig. 1, given dataset is fuzzy partitioned into overlapping clusters using a fuzzy clustering algorithm. In sequence, for each cluster identified, $i = 1,\ldots,c$, we predict a fuzzy function $\tilde{f}_i(x,\theta_i) \in \Re$ using a non-generative algorithm, e.g., regression, etc. To predict an output value of a new testing data point, $x^{test} \in \Re^{nv}$, we first use fuzzy clustering parameters to estimate membership values to each cluster and then using estimated fuzzy function parameters, $\hat{\theta}_{i=1,\ldots,c} \in \Re^{nv}$ we obtain a scalar output value, $\hat{y}_i = \hat{f}(x^{test},\hat{\theta}_i) \in \Re$. The arithmetic mean of the estimated output values for each cluster, which are weighted with their cluster membership values is calculated to obtain a crisp output value for the given test data point. We can further simplify the learning algorithm demonstrated in Fig. 1 in two steps:

(i) *Fuzzy clustering:* The domain $X \in \Re^{n\times nv}$ of nv dimensional input space is partitioned into c overlapping clusters using fuzzy clustering algorithm based on a chosen degree of fuzziness constant, m, and each cluster is represented with cluster centers, $V_i \in \Re^{nv}$ $i = 1,\ldots,c$, and membership value matrix, $U \in [0,1]^{n\times c}$, $U_{i,j} = \mu_{i,j} \in \Re$ where $i = 1,\ldots c$, $j = 1,\ldots,n$.
(ii) *Function approximation:* To each of these regions a local fuzzy model $\tilde{f}_i(x,\theta_i) \in \Re$ is calculated by using membership values, $\mu_i \in \Re^n$ as additional predictors to given input variables, $X = \{x_{k=1,\ldots,n}\}$, $x_k \in \Re^{nv}$. Scalar valued fuzzy functions are calculated by using any regression method, e.g., linear least squares, kernel regression, etc.

It should be pointed out that the consequent functions, which are called the fuzzy functions in this paper, are special functions which are formed using not only the original input variables but also the membership values of the particular cluster (rule) and its user defined transformations to improve the prediction performance. Thus we implement a special hybrid clustering method – IFC [8] – which can shape membership values so they can improve the representation of local dependencies by minimizing the local loss function, i.e., $\min \sum_i l(\theta_i) | l(\theta_i) = (y - \hat{f}_i(x,\mu_i;\theta_i))^2$.

Let (x_k, y_k) denote each training data point, where $x_k \in \Re^{nv}$ is any kth input vector of nv dimensions, $y_k \in \Re$ is its observed output value, $\mu_{ik} = \mu_{ik}(x_k) \in [0,1]$ represent its membership value to cluster $i = 1,\ldots,c$, c be the total number of clusters, $m > 1$ be the level of fuzziness parameter. The learning algorithm of type-1 fuzzy functions approach [8] is explained below.

2.2.1. Improved fuzzy clustering (IFC)

Improved Fuzzy Clustering (IFC) is a hybrid clustering method, which combines the point-wise clustering and function estimation methods in one objective function as follows:

$$\min J_m^{\mathrm{IFC}} = \overbrace{\sum_{i=1}^{c}\sum_{k=1}^{n}\mu_{ik}^{m}d_{ik}^{2}}^{\text{clustering-error}} + \overbrace{\sum_{i=1}^{c}\sum_{k=1}^{n}\mu_{ik}^{m}l_{ik}(y_k,g_{ik})}^{\text{function-approximation loss}} \tag{2}$$

0 In (2), $d_{ik} = \|x_k - v_i\|$, represents the distance from each x_k to each cluster center, v_i, and the first term measures the clustering error. We minimize this loss to stabilize the inter-cluster and inner-cluster similarities [19]. The loss function $l_i(\tau) = (y_k - g_i(\tau_{ik}))^2$ of the second term the squared deviation between approximated fuzzy models, namely the *interim* fuzzy

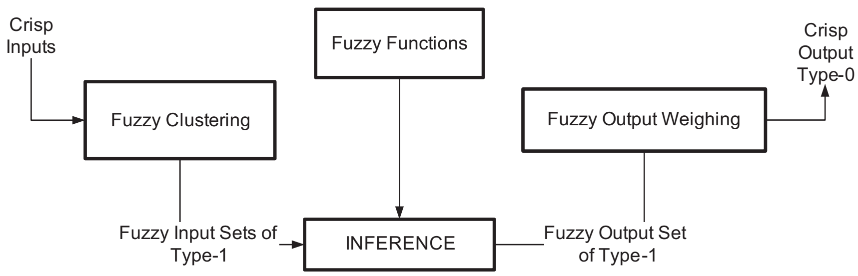

Fig. 1. Type-1 fuzzy functions systems.

functions $g_i(\boldsymbol{\tau}_i)$ of cluster i and the actual output. We would like to minimize this loss by estimating better interim fuzzy functions. The interim fuzzy functions can be estimated with any regression method, e.g., least squares estimation-LSE, multivariate non-linear regression, etc. At each step of the clustering algorithm, a different membership value is used as input variable. The aim is to capture the best membership's values as inputs that can explain the local input–output relationships and at the same time identify regression functions that define different dependencies between inputs and output. The first term in the objective function in (2) helps to identify the local regions and the second term helps to shape the membership values so the interim fuzzy function loss is minimized. The corresponding membership values and their possible transformations are the only predictors of the interim fuzzy functions $g_i(\boldsymbol{\tau}_i)$ excluding original variables. An example of an interim fuzzy function can be formed using:

$$g_i(\boldsymbol{\tau}_i) = \hat{w}_{0i} + \hat{w}_{1i}\mu_i + \hat{w}_{2i}(1 + \exp(-\mu_i^m)) \tag{3}$$

Additional examples of fuzzy functions can be found in [7,8]. In (3), $\hat{w}^i$ represents the regression coefficients. The second term of the objective function can be minimized if the optimum functions can be found. Thus, the algorithm searches for the best interim fuzzy functions, $g_i(\boldsymbol{\tau}_i)$. From the Lagrange transformation of the objective function in (2) the membership values are calculated as follows:

$$\mu_{ik} = \left(\sum_{j=1}^{c} \left[\left(d_{ik}^2 + l_{ik} \right) / \left(d_{jk}^2 + l_{jk} \right) \right]^{1/(m-1)} \right)^{-1}; \quad \sum_{i=1}^{c} \mu_{ik} = 1 \tag{4}$$

$i = 1,\ldots,c$, $k = 1,\ldots,n$. The cluster update equation is not affected and is same as the FCM [19] method. Punishing the objective function with an additional error forces the algorithm to capture membership values that would help to reduce the loss, but at the same time identify the hidden partitions. Membership function in (4) yields "*improved*" membership values, $\mu_{ik}^* \in [0, 1]$, such that the membership values improves the prediction of the local dependencies.

When estimating interim fuzzy functions, we eliminated the original input variables because we wanted to capture the membership values that can explain the output variable. Alienating membership values helps to shape them independent from the original input variables and prevent them from the effects of dominant input variables, which are highly correlated with the output. We demonstrate in Fig. 2 that the membership values obtained from IFC are better inputs to build a model to explain the output compared to the membership values obtained from the well-known FCM [19] method.

We constructed a very simple toy dataset of 20 data points with a single input and single output and executed FCM and IFC, respectively setting $c = 2$ and $m = 1.5$ for each clustering method. The top scatter diagrams in Fig. 2 are linear regression

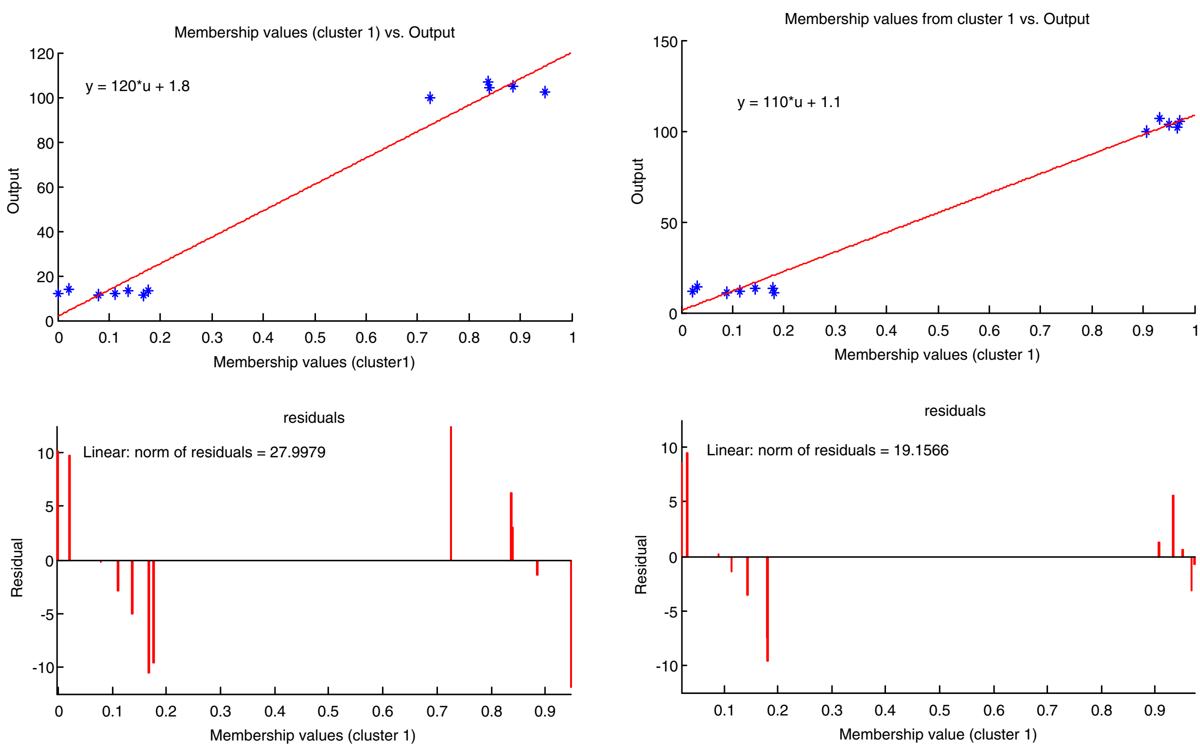

Fig. 2. Performance of membership values from FCM (left) and IFC (right). The top diagrams are functions of membership values from FCM and IFC to explain the output variable of a sample dataset. The lower diagrams are residual errors of corresponding linear functions on the top.

functions estimated using the membership values of a cluster as input variable. The top left figure demonstrates the membership values from FCM and the right from IFC. It is to be noted that the membership values of IFC (on the top-right of Fig. 2) can explain the output better and the residual error of a single regression function is less than that of the regression function estimated with the FCM results. The scatter diagram on the top-right shows that the membership values are shaped (aligned) close to the output variable. This is due the result of IFC, which can shape the membership values into explaining output variable. Different transformations of membership values, e.g., exponential or logistic, can help to explain the output variable.

2.2.2. Identification of local dependencies

One fuzzy function is approximated for each cluster to identify the input–output relations within each cluster i. The dataset of each cluster is comprised of original input variables, $\boldsymbol{x}$, improved membership values, μ^*_{ik}, of particular cluster i obtained from IFC, and their user defined transformations, e.g., $((\mu^*_{ik})^p\ (p > 1),\ e^{\mu*}$, etc. This is same as mapping nvdimensional input space, $\Re^{nv}$, of each individual cluster i onto a higher dimensional feature space $\Re^{nv+nm}$, i.e., $x \to \Phi_i(x, \mu^*_i)$. Here nm is the total number of membership value transformations used to structure a system of *principle fuzzy functions*, $\hat{f}(\Phi_i)$, to determine the local relations of each cluster in $(nv + nm)$ space. A sample fuzzy function structure Φ_i using two different membership value transformations and the original inputs is as follows:

$$\Phi_i = \begin{bmatrix} (\mu^*_{i1}) & (e^{\mu^*_{i1}}) & x_{1,1} & \cdots & x_{nv,1} \\ \vdots & \vdots & \vdots & & \vdots \\ (\mu^*_{in}) & (e^{\mu^*_{in}}) & x_{1,n} & \cdots & x_{nv,n} \end{bmatrix}_{n\times(nm+nv)} \tag{5}$$
$$\tilde{f}(\Phi_i;\beta_i) \in \Re | \beta_{i,0}\mu^*_i + \beta_{i,1}\exp(\mu^*_i) + \beta_{i,2}x_1 + \cdots + \beta_{i,(nv+nm)}x_{nv}$$

The interim fuzzy functions, $g_i(\boldsymbol{\tau}_i)$ are different from principle fuzzy functions $\tilde{f}(\Phi_i;\beta_i)$, since $g_i(\boldsymbol{\tau}_i)$ is used to shape the membership values during IFC and only use membership values and their transformations as input variables. Whereas principle fuzzy functions can minimize the local loss, $\min\sum_i l(\theta_i) | l(\theta_i) = (y - \hat{f}_i(x, \mu_i;\theta_i))^2$, i.e., the error between the actual output and the model output from each cluster induced by the improved membership values, μ^*_i. In sequence, we first obtain improved membership values, without the effect of the input variables, and then use them to calculate the fuzzy functions.

3. Interval-valued fuzzy functions (IVFF) approach

One of the challenges of type-1 fuzzy functions strategies summarized above is that the learning parameters of the clustering, e.g. the number of clusters, c, degree of fuzziness, m, and the parameters of the fuzzy functions, e.g., the definitions of fuzzy functions identified with different types of transformations, τ_i and Φ_i, are uncertain. One needs to define these parameters manually prior to model construction. Even if optimization methods such as genetic algorithms [15,18] or neural networks [16,20] are used to build hybrid fuzzy models, assigning a single (optimum) value to any of these imprecise parameters and generalizing it for every object of a given dataset may not reveal an optimum solution. An alternative way would be assigning interval-valued parameter values instead of crisp values, and precisiating such intervals during inference. Such a rich representation of a fuzzy model would improve performance, i.e., the optimum model would present a full solution to choose from among alternative best solutions. In this sense, the uncertainty modeling of this paper is based on identification of interval values for imprecise learning parameters.

Using interval-valued parameters to form interval-valued type-2 fuzzy systems has been studied by different researchers. In [10], Turksen proposed characterization of uncertainty of fuzzy systems by identifying upper and lower boundaries of the fuzziness parameter of the fuzzy c-Means (FCM) [19] clustering algorithm. The degree of *fuzziness*, m, viz., a constant to represent overlapping degree of identified clusters, has been investigated by many researchers, e.g., [3,11,21], since it is shown that changing the fuzziness parameter results in different membership values, which is a natural consequence of identification of interval-valued membership values. In addition, for different membership value transformations different fuzzy functions can be obtained, resulting interval-valued outputs for each cluster. Hence, in this paper, the new IVFF uses the fuzziness parameter of improved fuzzy clustering (IFC) and different structures of fuzzy functions to identify embedded membership values and embedded local scalar fuzzy functions. In particular, embedded interval membership are formed by using different values of fuzziness parameter (m) of IFC clustering [8]. Thus, we define the footprint of uncertainty (FOU) of interval membership values based on a fuzzy clustering parameter and function definitions. Next, we will present the architecture of the proposed method in two steps: structure identification and inference engines.

3.1. Structure identification of the proposed IVFF

To build ICFF, we use interval-valued membership values as shown in Fig. 3, thus a definition of an interval fuzzy set is required. Interval fuzzy sets, $\widetilde{A}$, map the domain of a variable onto membership values in the interval of [0,1] as follows:

$$\mu_{\widetilde{A}}(x) : x \in \left[\mu^L_{\widetilde{A}}(x), \mu^U_{\widetilde{A}}(x)\right], \quad \mu_{\widetilde{A}}(x) \in [0, 1] \tag{6}$$

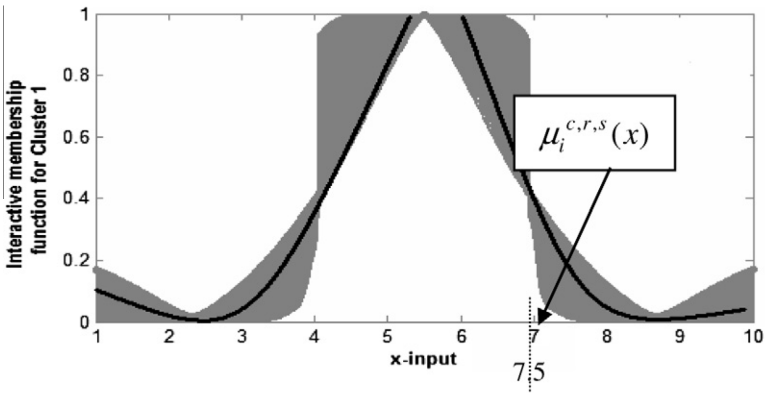

Fig. 3. Interval-valued membership values of a cluster formed by parameters of 3-tuples, $\langle c, m^r, \tau^s\rangle$. c is the number of clusters, m^r defines the level of fuzziness, τ^s represents the interim fuzzy function structures of the improved fuzzy clustering (IFC).

In (6) $\mu^L_{\tilde{A}}(x)$ and $\mu^U_{\tilde{A}}(x)$ are lower and upper membership functions. In computations, such interval is a precisiated interval defined based on different values of learning parameters as explained in the previous section. If the primary membership values of an interval-valued membership value distribution can be characterized by a set of parameters $p = (p_1, p_2, \ldots, p_{np})$ each taking discrete values and P is the domain, then interval-valued type-2 membership distribution such as in Fig. 3 is the collection of primary membership values determined for each possible value of $p^r \in P$. Therefore, the membership value distribution of a discrete interval-valued fuzzy set $\widetilde{A}$ can be reformulated as follows:

$$\mu_{\tilde{A}}(x) = \{\mu^r_A(x)\}, \quad r = 1, \ldots, nr \tag{7}$$

In (7), $\mu^r_A(x)$ is the rth primary membership value of a data point x and $\mu_{\tilde{A}}(x)$ is the collection of all the primary membership values that define an interval and nr is the total number of possible values of p^M. In this paper, interval-valued membership values are defined by changing the parameter of type-1 fuzzy functions. Hence, we will define IVFF model based on embedded type-1 fuzzy function parameters as follows:

We define a list of fuzziness values, $m^r \in \{m^1, \ldots, m^n r\}$, each $m^r \in \Re > 1.0$, where nr is the total number of discrete values of fuzziness parameter, e.g., $m^r \in \{1.1, 1.3, \ldots, 3.5\}$. We also define a set of fuzzy function definitions, τ, for interim fuzzy functions. Such functions are formed by using different membership values transformations. Thus, let τ^s represent an interim fuzzy function definition, then discrete set of interim fuzzy function definitions are defined as $\tau^s \in \{\tau^1, \ldots, \tau^{nif}\}$, where nif is the total number of interim fuzzy-function definitions. Examples of different interim fuzzy functions can be identified with different transformations of membership values. It should be noted that a single IFC model uses a discrete fuzziness value, m^r and a single fuzzy function definition τ^s when predicting interim fuzzy functions. With the combination of set of discrete fuzziness values and set of discrete interim fuzzy function definitions, we can identify a list of IFC models. Hence, let $t$represent a single IFC model. Then, one can define $t = 1, \ldots, (nrif = nr \times nif)$ different embedded improved fuzzy clustering–IFC–models (Fig. 4). We represent each IFC model with tuple.[1] of $\langle c, m^r, \tau^s\rangle$, $r = 1, \ldots, nr$, $s = 1, \ldots, nif$.

In an analogical manner to discrete definition of IFC models, we define a set of principle fuzzy functions. We define different principle fuzzy functions by using different combinations of improved membership values as input variables. Let a single principle fuzzy function definition is given by Φ^ψ, where $\Phi_i(\boldsymbol{x}, \mu^*_i)$ is a representation of the input matrix with the use of membership values as input variables. Then a discrete set of function definitions (such as in (5)) can be defined for up to nfdifferent definitions, $\psi = 1, \ldots, nf$. We also define different structures for each cluster of a single model. Hence that; local fuzzy functions of two different clusters of a corresponding model may not need to be same, i.e., $\Phi^\psi_{i=1} \neq \Phi^\psi_{i=2}$. To give an example, for one cluster we could define fuzzy functions with only membership values as additional input variables and for another cluster we could define fuzzy functions with different transformations of membership values. Thus, each model would embed different function forms for each separate cluster.

Initially, the embedded IFC models will form interval membership values as described in the following:

3.1.1. Learning step 1: clustering with IFC

Here, we will explain how we can form interval-valued membership value distributions for each cluster. Each tuple $\langle c, m^r, \tau^s\rangle$ represented with $t(tuple)$ characterizes an IFC model. To construct interval-valued membership values (membership value distributions for the entire dataset), we construct $nrif$ different IFC models, $t = 1, \ldots, nrif$. As a result, each t IFC model identifies c number of cluster centers, represented with $\upsilon_i^{c,r,s}(x|y) = (\upsilon^{c,r,s}_{i,1}, \upsilon^{c,r,s}_{i,2}, \ldots, \upsilon^{c,r,s}_{i,nv}, \upsilon^{c,r,s}_{i,nv+1}) \in \Re^{nv+1}$ (since input–output dataset is used to find the cluster centers) and membership values, $\mu_i^{c,r,s}(x|y)$ are captured in input–output $(x|y)$ space.

[1] In this paper, **tuple** represent finite sequence of different parameters with no particular order. A tuple containing n objects is also represented with "n-tuple".

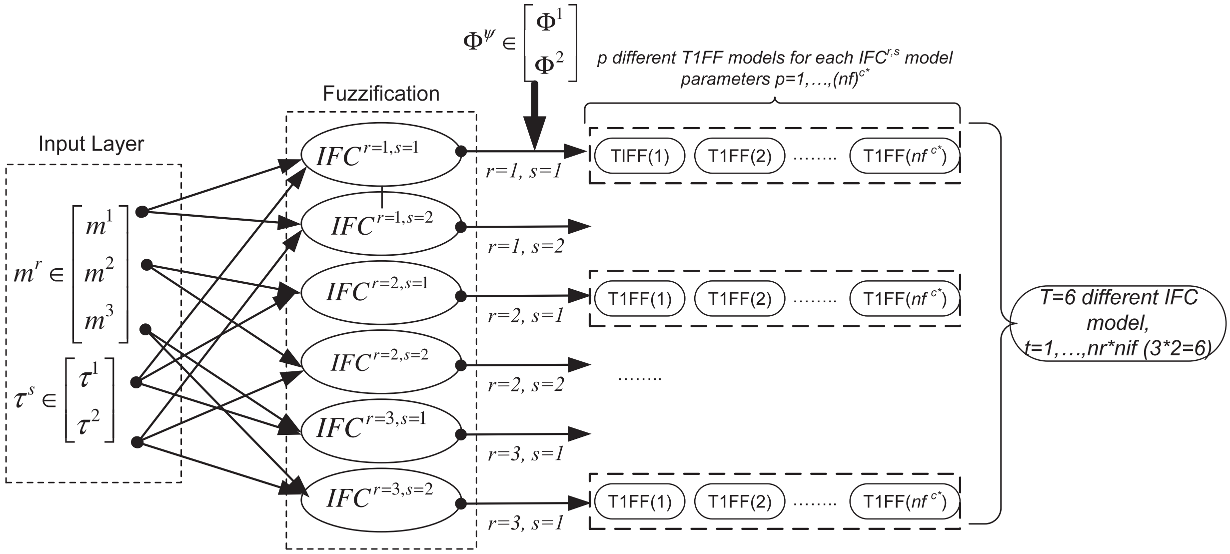

Fig. 4. The structure identification of the IVFF. The shaded fuzzy functions represent the fuzzy functions of the corresponding cluster. The output value of kth data point, y_k, is calculated using each output obtained from these fuzzy functions.

Thus, $\mu_i^{c,r,s}$ represents the ith cluster's membership values obtained from running the IFC with m^r parameter and τ^s interim fuzzy function definition.

Next, we map the $x|y$ membership values onto input domain to find the cluster centers of matrix $\boldsymbol{x}$, $\upsilon_i^{c,r,s}(x) = (\upsilon_{i,1}^{c,r,s}, \upsilon_{i,2}^{c,r,s}, \ldots, \upsilon_{i,nv}^{c,r,s})$ using the cluster center update equation which is same as FCM [19]. Membership values of input domain, to be used as input variables to obtain the cluster centers of the input domain, are calculated using Eq. (4). We plotted a sample input membership values of input domain in Fig. 3. Here, we demonstrate uncertainty interval of membership value distributions induced by changing values of m^r and τ^s in gray. Recall that these membership value distributions are idealized representations obtained after a curve-fitting over the scatter diagram of membership values.

The bounded interval in Fig. 3 comprises of discrete membership values (of a particular cluster i) one for each different parameter set obtained from Cartesian product of its discrete values ($nr \times nif$). Hence, for each datum x, *e.g.*, $x = 7.5$, $t = 1, \ldots, nr \times nif$ different membership values can be defined within this interval. Each one of these t membership values are represented with $\mu_i^{c,r,s}(x)$ associated with tth IFC model using tuples of $\langle c, m^r, \tau^s \rangle$.

If we define one membership value for each discrete data point x_k, we would form one discrete membership value distribution for the given cluster, as shown in black line in Fig. 3.

3.1.2. Learning step 2: approximation of embedded fuzzy functions

As shown in the previous subsection, we obtain interval-valued membership value distributions by identifying discrete interval-valued parameters of IFC model. Once improved interval-valued membership values of each cluster are obtained using IFC, using these improved membership values as additional input variables, we characterize different forms of principle fuzzy functions for each cluster. These fuzzy functions would characterize local models. We can define as many fuzzy functions for each cluster by introducing different transformations of membership values as input variables. Thus, let nf represent the set of different function definitions that we would use to define a local model. Then, we can identify nfdifferent fuzzy functions for each cluster i, each of them represented with $\Phi_i^{\psi},\ \psi = 1, \ldots, nf,\ i = 1, \ldots, c$. Thus, using the membership values of any IFC model from within the list of discrete membership values, viz. each set of $\langle c, m^r, \tau^s \rangle$ parameters, we can identify nf different principle fuzzy functions for each cluster, denoted with $f_i^{r,s,\psi},\ i = 1, \ldots c$. This corresponds to identification of embedded fuzzy functions.

The precisiation of fuzzy functions to represent the proposed IVFS can be better explained with an example as shown in Figs. 4 and 5. We define precisiated parameters at the start of the system modeling. Thus, let the following list of parameters are identified at the start of the algorithm: $m^r \in \{m^1, m^2, m^3\}$, $nr = 3$, $\tau^s \in \{\tau^1, \tau^2\}$, $nif = 2$ and $\Phi^{\psi} \in \{\Phi^1, \Phi^2\}$, $nf = 2$. In words, we define three ($nr = 3$) different discrete fuzziness values $m^r \in \Re > 1.0$, two ($nif = 2$) different interim fuzzy function definitions each of them are represented with τ^s and lastly two ($nf = 2$) different principle fuzzy function definitions each of which are represented with Φ^{ψ}.

Each τ^s represents one interim fuzzy function structure to be used to build one IFC model, a total of two different interim fuzzy function structures are determined here. It composes of the membership values and their transformations only. As a result,$\{(nr = 3) \times (nif = 2)\} = 6$ separate IFC models can be identified, $t = 1, \ldots, 6$, where each discrete IFC model is denoted with t. At this stage, the optimum number of clusters, c^*, is also a given parameter. Each Φ_i^{ψ} represents one (system) fuzzy

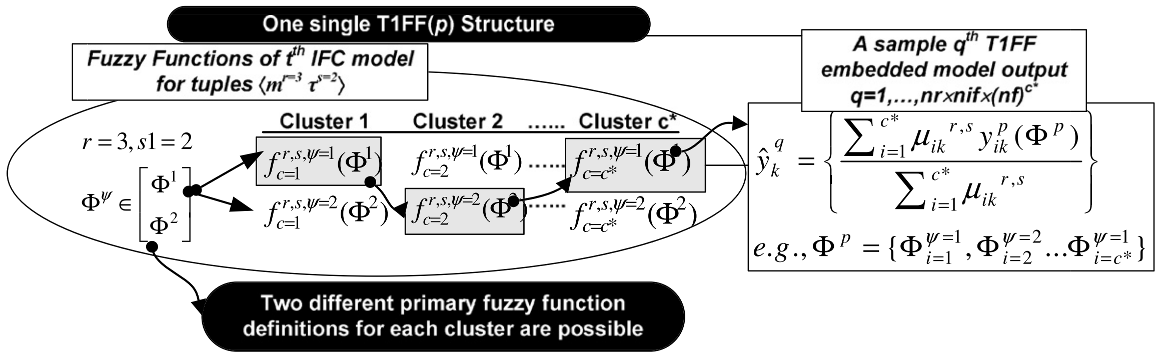

Fig. 5. Identification of the parameters of the optimum embedded models from among set of discrete Type-1 Fuzzy Functions (T1FF) for any training input vector, x_k, $k = 1,\ldots,n$.

function definition $f_i^{r,s,\psi}$ to be used to identify local principle fuzzy functions of each cluster. For each tth IFC model, $t = 1,\ldots,nif \times nr$, e.g., in our example 6 different IFC models can be build, nf different local fuzzy functions $f_i^{r,s}(\Phi_i^{\psi})$ can be approximated for each cluster $i,\psi = 1,\ldots,nf$. If there are c^* number of clusters and for each cluster nf different fuzzy functions can be identified, then there can be $p = 1,\ldots,(nf)^{c^*}$ different embedded fuzzy functions for each single IFC model.

Each of the $(nf)^{c^*}$ models would define a single type-1 fuzzy function model (T1FF). A combination of each T1FF would yield the proposed IVFF structure. In the example above, there are two different fuzzy function structures to choose from to identify interim fuzzy functions $g_i(\tau)$ as well as principle fuzzy functions of each cluster. We also use pre-determined α-cut > 0 value to eliminate some data points $\mu_{ik} < \alpha$-cut that do not affect the decision surfaces of a corresponding cluster.

In different clusters, different fuzzy function models may produce better results based on a pre-defined performance measure. For instance, one specific model can reduce the error better than the others for a specific $m$value. In another cluster, different fuzzy functions for different fuzziness levels could be more preferable. One needs to determine the best fuzzy function definitions for each cluster separately. Therefore, we estimate output values from each embedded model and choose the optimum models for each training vector separately as shown in detail in Fig. 5. Here $\Phi^p = \{\Phi_1^1, \Phi_2^2, \ldots, \Phi_{c*}^1\}$ represents the definition of one embedded type-1 fuzzy function model structure obtained using tth IFC model outputs, e.g., membership values and parameters. Each Φ_i^{ψ} in Φ^p represents a fuzzy function definition denoted with ψ, $\psi = 1,\ldots,nf$, for a cluster i. Here $\hat{y}_k^q$ is a single estimated output value for kth input vector obtained from a single embedded model. The aim is to keep the parameters of the optimum model for each training data point separately that would minimize the model‘s overall loss, i.e., $\min \sum_k l_k | l_k = (y_k - \hat{y}_k^q)^2$. After we identify the best model parameters for each training sample separately, we retain them in a database, i.e., matrixes entitled m-collection, τ-collection and Φ-collection tables, which include optimum parameters of each training vector. These collection tables can be constructed in the following way:

Hence, let $\hat{y}_k^q,\ q = 1,\ldots,nif \times nr \times (nf)^{c*}$, be embedded model output value of kth data point using different fuzzy functions $f_i^{r,s,\psi}$ of each cluster i, the fuzzy function structures of qth embedded model is represented by $\Phi^p = \{\Phi_{i=1}^{\psi}, \ldots, \Phi_{i=c*}^{\psi}\}$. Φ_i^{ψ} represents each local fuzzy function structure, $\psi = 1,\ldots,nf$. One fuzzy function, $f(\Phi_i^{\psi})$ or $f(\Phi_i^{r,s,\psi})$ is approximated for each cluster i for each set of IFC models represented with $\langle c^*, m^r, \tau^s\rangle$ to approximate output values for each vector k. The model output $\hat{y}_k^q$ is calculated by fuzzy weighted average method using tuples $\langle c^*, m^r, \tau^s, \Phi^p\rangle$ as follows:

$$\begin{aligned} \hat{y}_k^q &= \left\{ \frac{\sum_{i=1}^{c*} \mu_{ik}^{r,s} y_{ik}^{r,s,\psi}(\Phi_i^{\psi})}{\sum_{i=1}^{c*} \mu_{ik}^{r,s}} \right\} \\ \Phi^p &= \{\Phi_{i=1}^{\psi=1}, \Phi_{i=2}^{\psi=2}, \ldots, \Phi_{i=c*}^{\psi=1}\} \end{aligned} \tag{8}$$

In (8) $y_{ik}^{r,s,\psi}(\Phi_i^{\psi})$ represents each model output of kth data vector in ith cluster obtained from the chosen fuzzy functions of each cluster $f_i(\Phi_i^{r,s,\psi})$. Thus, for each embedded fuzzy function model of a IVFF system, represented with q, $q = 1,\ldots,nif \times nr \times (nf)^{c^*}$, one output value for each kth data point, $\hat{y}_k^q$, is obtained, $k = 1,\ldots,n$.

One can identify the optimum model output from among all embedded models, which would give the minimum loss. Hence, for each data point k in the dataset, we measure the squared error between the actual and predicted output obtained from each qth embedded model, each of which is represented with tuples $\langle c^*, m^r, \tau^s, \Phi^p\rangle,\ \Phi^p := \{\Phi_i^{\psi}\},\ p = 1,\ldots,(nf)^{c^*},\ \psi = 1,\ldots,nf,\ i = 1,\ldots,c^*$. Then the optimum parameter set for each data point k is measured with the following equation:

$$\arg min l(q) = \arg\min_q (y_k - \hat{y}_k^q)^2 \in \left\{ q | \exists q', (y_k - \hat{y}_k^q)^2 < \left(y_k - \hat{y}_k^{q'}\right)^2 \right\} \tag{9}$$

As a result, for each kth training vector, the optimum output $\hat{y}_k^q$ among $q = 1,\ldots,nif \times nr \times (nf)^{c^*}$ different embedded models is selected. The parameters of this chosen optimum model are retained in collection tables for each kth data point as follows ($k = 1,\ldots,n$):

$$mCol^{n\times 1} = \begin{bmatrix} m_1 \\ \vdots \\ m_n \end{bmatrix}, \quad \tau Col^{n\times 1} = \begin{bmatrix} (\tau_1) \\ \vdots \\ (\tau_n) \end{bmatrix}, \quad \Phi Col^{n\times c*} = \begin{bmatrix} (\Phi_{1,1}) & \cdots & (\Phi_{c*,1}) \\ \vdots & \ddots & \vdots \\ (\Phi_{1,n}) & \cdots & (\Phi_{c*,n}) \end{bmatrix} \tag{10}$$

Next, we present the proposed inference method. In (10) the optimum parameters of a training vector k would be the kth row of m-Col, τ-Col, Φ.

3.2. Inference engine of the proposed IVFF

During the structure identification step of the algorithm, discrete interval-valued membership values and fuzzy functions are identified. Each combination of parameters represents a single embedded fuzzy type-1 fuzzy function model. During inference method, we start with the reduction of the type of the interval-valued membership values down to one single membership value using a "case-based type reduction" method. Fig. 6 depicts the steps of the proposed inference structure.

3.2.1. Inference step 1: cased-based type reduction

Cased-based reasoning (CBR), which is a novel Artificial Intelligence (AI) problem solving paradigm, involves adaptation of old solutions to meet new demands, explanation of new solutions using old instances (called cases), and performance of reasoning from precedence to interpret new problems. It has a significant role in today's pattern recognition and data-mining applications. In this work, this phenomenon is implemented to reduce the interval-valued fuzzy sets down to type-1 fuzzy sets, namely, the *type-reduction* method.

In the standard interval-valued type-2 fuzzy logic systems, the first step is to identify the interval membership values. Then type-2 operations are applied on these interval-valued type-2 fuzzy sets to construct aggregation and implication operations. Then a type reducer is applied on the output fuzzy set first to reduce the type-2 to type-1 and then defuzzification to find a crisp output value, viz. reducing the type-1 to type 0. In the new inference method, since the output fuzzy sets are represented with fuzzy functions, but not with output fuzzy sets, defuzzification method is not required. However, we do need a type-reduction to reduce the interval-valued fuzzy sets down to type-1. Instead of working with the interval-valued type-2 membership values during inference, we apply the new case-based type-reduction method, right at the beginning of the algorithm and work with type-1 fuzzy sets throughout the inference mechanism. It is this concept of the new inference mechanism that separates it from the earlier type-2 inference methods. After this point, the computations with type-1 fuzzy sets will be much easier than the type-2 fuzzy sets (since it is much easier to do operation with type-1 fuzzy sets which are crisp values not intervals). The discrete nature of fuzzy sets and fuzzy functions enable easy type reduction. The steps of the new inference methodology are sketched in Fig. 7.

It should be noted that a unique property of the IVFF strategy is that, the uncertainty is not only identified by constructing interval-valued membership values, but also constructing interval-valued fuzzy functions. The optimum fuzzy function of each local model is uncertain. Thus we approximate a list of fuzzy functions based on different forms of membership values, to define a footprint of uncertainty for each cluster. Each data point k would have a number of different crisp output values when different fuzzy functions are used. The aim is to select the optimum local fuzzy function which would be the best fit to the given data vector. There could be unlimited number of functions, but we define only a list of (embedded) fuzzy functions within an uncertainty interval by discretization of the uncertainty interval to allow easy computations with type-1 fuzzy functions and membership values. This structure is depicted in Fig. 7.

Given the test data vectors with unknown output values (x^{test}) and the training input data vectors with known outputs, (x_k, y_k), $k = 1,\ldots,n$ and their corresponding collection tables, $\{mCol^{n\times 1}, \tau Col^{n\times 1}, \Phi Col^{n\times c^*}\}$. For each lth testing vector we execute the following.

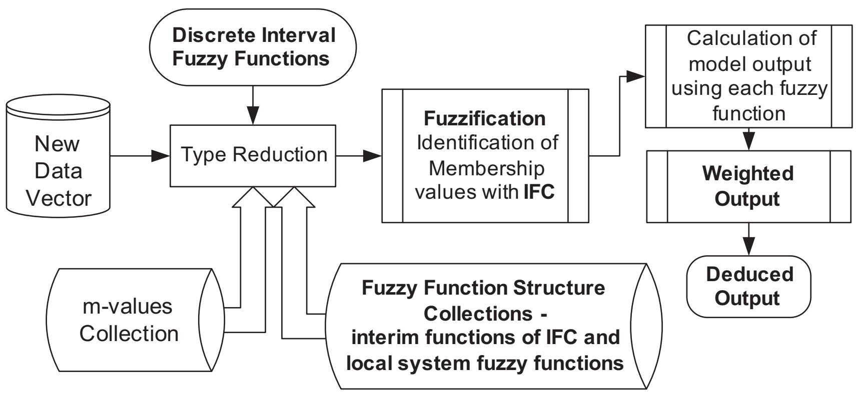

Fig. 6. IVFF inference module framework.

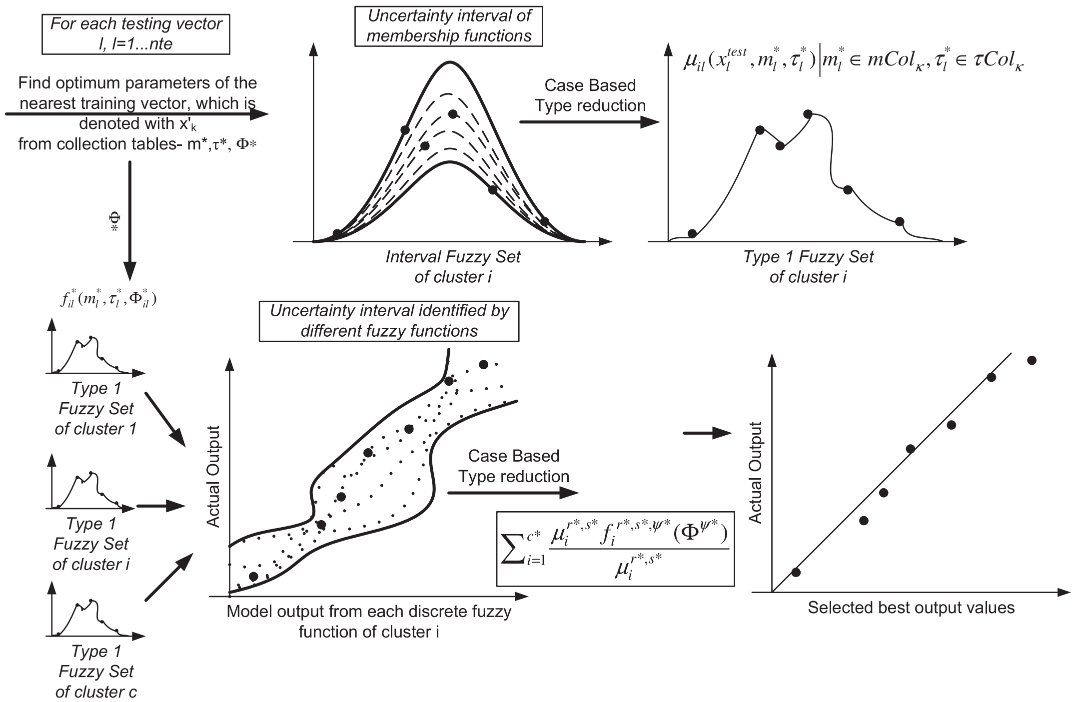

Fig. 7. Cased-based type reduction for inference method of IVFF system. The dark dots (•) represent the optimum membership values and fuzzy functions determined for each testing vector, l, $l = 1,\ldots,$nte.

As the first step of the case-based type reduction method, we identify the nearest training vector k' to the lth testing vector using Euclidean distance measure as follows:

$$\arg\min_{x_{k^*}} d(x_l^{test}, x) \in \{l | \exists x_k, x_{k'} d(x_l^{test}, x_{k'}) < d(x_l^{test}, x_k), \quad k, k' = 1, \ldots, n\} \tag{11}$$

The optimum parameters, indicated with a (*), to identify membership values and the structure of each fuzzy function in each cluster, i, $i = 1,\ldots,c^*$ are captured from collection tables based on k' follows:

$$\begin{aligned} &\mu_{il}^{imp}(x_l^{test}, m_l^*, \tau_l^*) | m_l^* \in mCol_{k'}, \quad \tau_l^* \in \tau Col_{k'} \\ &f_{il}(m_l^*, \tau_l^* | \Phi_{ik'}^*) | \Phi_{il}^* \in \Phi_i Col_{k'} \end{aligned} \tag{12}$$

In (12) m^* represents the optimum level of fuzziness identified from the m-Col, *i.e.*, collection table, for lth test vector x_l^{test} based on the closest training vector $x_{k'}$. Similarly, τ_l^* represents the optimum interim fuzzy function structure identified specifically for x_l^{test} from the τ-Col table based on $x_{k'}$ and it holds the information about the membership values and their transformations and their regression coefficients. In analogical form, Φ_{il}^* holds the optimum local fuzzy function structures, one for each cluster, $i = 1,\ldots,c^*$, identified specifically for x_l^{test} using the Φ-Col table based on $x_{k'}$ and it holds the information about the membership values and their transformations and input variables and their regression coefficients.

The optimum parameters to do reasoning is determined based on the optimum parameters of the nearest training vector, $x_{k'}$, from the collection tables. Then for each testing vector, x_l, the optimum embedded models are identified based on the case-based type reduction method.

3.2.2. Inference step 2: fuzzification

In (12), m_l^* and τ_l^* represent the optimum values of degree of fuzziness and interim fuzzy function parameters of the testing vector, l. The membership values of the lth testing vector is measured using IFC membership value calculation equation in (4), using the optimum parameter tuple, $\langle c^*, m_l^*, \tau_l^* \rangle$.

In the first step, κ training data samples that are nearest to the lth testing data sample, $l = 1,\ldots,nte$, are identified based on the Euclidean distance measure. Using interim fuzzy function parameters, $\{\tau_l^*, \hat{w}_l^*\}$, membership values of κ-nearest training data samples are calculated and κ different vectors are formed to build the interim matrix of κ data vectors based on the membership value transformation identified by τ_l^*. As a result an interim matrix for each cluster $i, \tau_i^* = [\tau_{i1}^* \cdots \tau_{ik*}^*]^T$ which compose of κ vectors are formed. This interim matrix is used to estimate the membership values of the corresponding testing sample, l. Then using the interim matrix structure, $\tau_i^*,\ i = 1,\ldots,c$, IFC model outputs of each κ-nearest training sample is obtained, $g_i(\tau_{iq}^*, \hat{w}_{il}^*), q = 1\ldots\kappa,\ i = 1,\ldots,c^*$.

The next step is to measure squared error values using the actual and model output values obtained from the interim fuzzy functions for each cluster using these nearest κ data samples, i.e., $SE_{iq} = (y_q^* - g_i(\tau_{iq}^*, \hat{w}_{il}^*))^2$, $q = 1,\ldots,\kappa$, $i = 1,\ldots,c^*$, to be used to approximate the average SE_i for the lth test data sample.

Next, error values, SE_{iq}, are weighted with weight constants, η_{rq}, which are normalized distances of the κ -training samples to testing sample l. The average approximate squared error of the lth testing sample in ith cluster is calculated with weighted square error, $_{m}\ \mathrm{SE}_{il}$, which is used in the new membership function to calculate improved membership values of the testing samples, μ_{il}. The next step is finding the fuzzy model outputs.

3.2.3. Inference step 3: identification of fuzzy model outputs

The fuzzy model output value is calculated for the lth testing vector, one for each cluster i, $i = 1,\ldots,c^*$, using linear functions such as LSE. The optimum parameters of the input matrix, Φ_{il}^*, are captured from the collection table $\{\Phi\mathrm{Col}_l^{n\times c*}\}$ during the case-based type reduction step of the algorithm. Using these optimum parameters, the fuzzy model outputs are calculated for each testing vector l by using optimum fuzzy function parameters as follows:

$$\begin{aligned} &\hat{y}_{1,l} = f_1(\Phi_1^*) : x_l^{test} \rightarrow \Phi_{1l}^*(x_l^{test}, m_l^*, \tau_l^*) \\ &\vdots \\ &\hat{y}_{c*,l} = f_{c*}(\Phi_{c*}^*) : x_l^{test} \rightarrow \Phi_{c*l}^*(x_l^{test}, m_l^*, \tau_l^*) \end{aligned} \tag{13}$$

Single crisp output value for lth testing vector is obtained by weighted fuzzy output method;

$$\hat{y}_l = \frac{\sum_{i=1}^{c*} \mu_{il}\hat{y}_{i,l}}{\sum_{i=1}^{c*} \mu_{il}} \tag{14}$$

In (14) μ_{il} represents the improved membership values calculated for lth testing vector in cluster i.

There are many structural differences between the new inference method (Figs. 6 and 7) and the inference methods of earlier approaches, e.g., [11,3]. To begin with, the new method is based on the interval-valued fuzzy functions approach; the earlier methods are based on type-2 fuzzy inference systems. The new IVFF approach replaces the implication and aggregation of the output fuzzy sets of the earlier inference methods with one single step, as described in*Inference Step 3*.This is due to the fact that the new method does not utilize fuzzy rule base (FRB) operations during inference. Type reduction is processed as the first step, and it is based on a search process, as opposed to earlier type-2 fuzzy systems in [1], which requires more extensive fuzzy operators on type-2 fuzzy sets. The type reduction is based on simple case-based reasoning and no defuzzification operation is required.

4. Application of the proposed IVFF method on desulphurization process of the steel industry

4.1. Desulphurization process

Desulphurization is a sub-process of a hot metal pre-treatment process, which takes place in a primary metal production industry. Hot metal produced in blast furnaces contains impurities like phosphorus, silicon, sulphur, carbon, and so on. The process of removing the impurities is called the refining process. The refining process consists of hot metal pre-treatment conducted in a torpedo car, decarburization process in a converter and various kinds of secondary refining processes corresponding to the requirements of final product. The hot metal pre-treatment sequential process is shown in Fig. 8. The major part of pre-treatment is assigned to desulphurization process.

The aim of the data-mining project is to build a decision support system to determine the right amounts of the reagents to be added into the hot metal. In reality, the target amount of sulphur (i.e. the aim sulphur) is often set much lower than the true sulphur value in desulphurization process. The argument underlying the modeling exercise is that a reduction in reagent consumption would be possible if more precise and reliable model can be developed to estimate the right amount

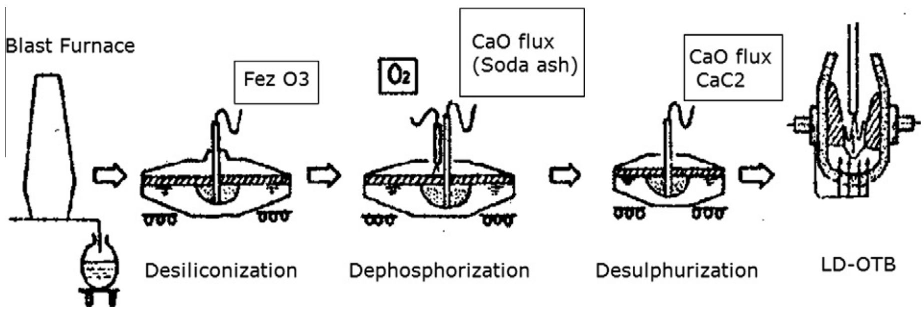

Fig. 8. Process flow of hot metal pre-treatment.

of reagents used in the desulphurization process. The Aim-Sulphur is the required quantity demanded by the customers of the steel plant. Empirical modeling strategy is required to understand the mechanism of the chemical and mechanical effects. One of the key issues of this modeling case study is that, when a model with poor predictive capability is used, it results in many batches of hot metal that has to enter desulphurization process again. It should be noted that the desulphurization process is a highly expensive process; therefore, the main objective in this modeling approach is to minimize the number of desulphurization processes increasing the modeling prediction ability.

4.2. Dataset

The desulphurization dataset consisted of approximately 13,000 data vectors with 27 attributes composed of binary, ordinal, scalar, and a few categorical variables. Each vector represents the measurements taken from one batch of a hot metal. Several variable selection methods are applied to choose the optimum parameters. Around 3000 data vectors in the dataset contained negative Reagent 1 and Reagent 2 values. Based on the information obtained from the domain experts in Dofasco, these vectors are considered as faulty inputs, and should be discarded from the dataset. Therefore, the number of vectors in the core dataset is reduced down to 10,000 data vectors just for this study (see Table 1).

We also applied statistical methods to clean the dataset from possible noises. In order to apply noise cleansing based on statistical methods, the probability distributions are drawn and the vectors that are outside a certain confidence value, e.g., $x_k > (5^{*}$ standard deviation) are discarded. These values are determined by the experts. The number of input vectors, after all of the above outlier treatments, has dropped to 9675 observations.

The dataset has two output variables; *Reagent 1* and *Reagent 2*. Therefore, for each output, we build separate models. The dataset is randomly divided into 1675 for training and validation purposes (500 for training and 250 for the validation) and 8000 observations are used to test the optimum model performance. Experiments were repeated with five random subsets of training and testing datasets of the above sizes. The R^2 values are used to measure performance of the models by averaging error rates of testing samples across five repetitions for each model.

4.3. Benchmark analysis

In this section, we present the experimental analysis conducted to investigate the performance of the proposed IVFF approach in comparison to different type-1 fuzzy inference systems e.g., adaptive neuro-fuzzy inference system ANFIS [16], dynamic evolving neuro-fuzzy inference system DENFIS [12], and type-2 fuzzy system entitled discrete interval type-2 fuzzy inference system DIT2FIS [3] are used. Non-fuzzy soft-computing methods, such as multilayer perceptron neural network (NN) [20], and support vector machines (SVM) [22] are also used as benchmark methods. In Table 2, we list the parameters used in these models.

The parameters of the optimum models of the IVFF method is retained in *collection tables*, i.e., $m\text{-}Col^{*}$, $\tau\text{-}Col^{*}$, $\Phi\text{-}Col^{*}$. There is one collection table set, i.e., $\langle m\text{-}Col^{*}, \tau\text{-}Col^{*}, \Phi\text{-}Col^{*}\rangle$, for each cross validation iteration, i.e., there are five different collection table sets obtained from each five different training and validation datasets of each Reagent 1 and Reagent 2 models, in order to do inference on five different testing datasets. The parameters of the optimum IVFF methodology from five different cross validation models are shown in Table 3.

We used the *r*-square (R^2) as a performance measure. The numbers in each cell in Table 4 represent the average R^2 values on testing datasets, from cross validation experiments. The values in parenthesis indicate the standard deviation of R^2 over five iterations. The lower the standard deviation the robust the applied method would be. Based on the results shown in Table 4, the proposed IVFF outperforms the rest of the models in terms of R^2 performance measure. For Reagent 1 and Reagent 2 datasets the R^2 differences between the best benchmark model, SVM, is around 2%.

In order to strengthen our findings, we further analyzed significance of the improvements. The *t*-test results of Reagent 1 and 2 models are as follows:

Table 1
Desulphurization dataset variables.

Variable name	Descriptions
Start-Sulphur	Starting level of sulphur before desulphurization
KGS	Weight of the batch that consists of iron (tons)
TEMP	Temperature of the hot metal as it leaves from the Blast Furnace
FB	Measure of fullness of the furnace
Aim-Sulphur	The amount of sulphur that is targeted to remain after desulphurization
End-Sulphur	The amount of sulphur remain within the metal after desulphurization
Compound 1–5	The chemicals measured in the hot metal as they arrive the desulphurization process
Car-Type	Specific style of vessel that is used to hold the hot metal
POS	Specific station at which the desulphurization takes place
Practice 1–6	They indicate that a certain type of modification to the normal operating practice has been applied
Injection Number	Number of times the hot metal goes through desulphurization process
Equipment Type	Equipment style used for the corresponding batch

Table 2
List of parameters values used in the experiments

Learning parameters
SVM-Regression [22]: $Creg \in [2^{-3}, 2^7]$, $epsilon \in (0.0, 0.5]$, two different kernel functions, i.e., linear $K(x_k, x_j) = x_k^T x_j$, or, non-linear Gaussian radial basis kernel (RBF), $K(x_k, x_j) = \exp(-\delta\|x_k - x_j\|)$, $\delta > 0$
Neural Network [20]: multilayer feed-forward structure with back-propagation optimization, hidden layer hyperbolic tangent sigmoid function and outer layer is a linear transfer function
ANFIS [16]: hybrid method to optimize inference parameters, Gaussian input membership function shapes, linear rule base structure (TSK)
DENFIS [12]: Takagi–Sugeno online training
DIT2FIS [3]: $c = [2, 10]$, m-*bounds* = [1.01, 3.5], TSK structure
Proposed IVFF: SVM is used to approximate the fuzzy functions. *Creg_bounds* = $[2^{-3}, 2^7]$, epsilon_bounds = (0.0, 0.5], m-bounds = [1.01, 3.5], c = [2,10], $\{\tau, \Phi\} = \{(\mu_i^p)\ p > 0;\ \exp(\mu_i^p);\ \ln((1 - (\mu_i)/(\mu_i))\}$

Table 3
Optimum parameters of the IVFF methodology obtained from cross validation trials.

Model name	IVFF
Fuzzy clustering type	Improved fuzzy clustering
Regression Type	Non-linear SVM
# of clusters	{6, 7, 8}
Fuzziness degree	[1.23, 1.89]
Optimum List of membership value transformations to be used as additional input variables	(μ), (e^{μ}), $(\log(1-\mu)/\mu)$
κ (number of nearest training vectors for IFC)	{2}
Alpha-cut	[0, 0.1]
C-regularization	{1.26, 5.25, 10.8, 103}
Epsilon	{0.03, 0.09, 0.23}
m-Col, τ*-Col*, Φ*-Col tables*	

Table 4
R^2 performance measures of the proposed and benchmark methods on testing dataset.

Modeling method/dataset	*Reagent 1*	*Reagent 2*
SVM-regression	0.789 (0.011)	0.776 (0.024)
NN	0.767 (0.010)	0.774 (0.014)
ANFIS (type-1)	0.591 (0.051)	0.624 (0.07)
DENFIS (type-1)	0.686 (0.029)	0.686 (0.005)
DIT2FIS (type-2)	0.745 (0.008)	0.751 (0.01)
Proposed IVFF	**0.805** (0.005)	**0.807 (0.005)**

Table 5
Two-sample left tailed t-test results ($p < 0.05$) for Reagent 1 and Reagent 2 Datasets. FR: Fail to Reject the Null Hypothesis, R: Reject the Null-Hypothesis. The numbers below each decision indicate the probability of observing the decision (FR/R).

	ANFIS	DENFIS	NN	SVM	DIT2FRB
Reagent 1					
IVFF	FR	FR	FR	FR	FR
	1	1	0.99	0.07	1
Reagent 2					
IVFF	FR	FR	FR	FR	FR
	1	1	0.07	0.11	0.97

The null hypothesis of t-tests for Reagent 1 experiments indicates that the performances of two paired algorithms are significantly different. The null hypothesis for Reagent 1 and Reagent 2 models is as follows:

$$H_0 : \left(\frac{1}{5}\sum_{cv=1}^{5} R^2_{j,cv}\right) - \left(\frac{1}{5}\sum_{cv=1}^{5} R^2_{k,cv}\right) > 2.5\%$$

The H_0 indicates that the difference between the average R^2 values obtained from five cross validation models of methodology j (row) and k (column) is greater than 2.5% (0.025 points in R^2 value). Each cell entry should be interpreted as follows: "*the methodology in the row is (not) significantly better than the methodology in the column*". Failing to Reject (FR) the null hypothesis indicates that the null hypothesis is true at the 95% confidence level and the methodology *j* is significantly better than the algorithm *k*. Rejecting the null hypothesis indicates that the two methodologies are not significantly different. For instance, as can be seen from the last row of Table 5, the optimum models of the proposed IVFF method are significantly different from the benchmark methods since the null hypothesis is not rejected.

5. Conclusions

In this study, a new interval-valued fuzzy inference system using improved fuzzy functions is presented. Structurally, the novel fuzzy system modeling structure is different than traditional fuzzy rule base approaches. The new method employs a new method to identify interval-valued fuzzy sets and fuzzy functions at a granular level. Two different types of uncertainties, namely the uncertainty in selection of improved fuzzy clustering parameter, and uncertainty in determining the mathematical model structure of each local fuzzy function, are taken into consideration. Additionally, earlier computationally efficient inference mechanisms are extended to incorporate aforementioned types of uncertainties during inference method. The proposed system modeling approach is applied on a steel production process to build a decision support tool. The cross validation error comparison analysis with earlier type 1 and type 2 fuzzy inference systems indicate that the proposed model is more robust with less error.

With the implementation of the proposed decision support tool on the steel production process, the steel company and the consumer will benefit as a result of reduced excess use of expensive material due to ineffective models. Increasing the efficiency in massive industrial processes will have an extraordinary positive effect on the environment.

Acknowledgement

This research is funded by NSERC – Natural Sciences and Engineering Research Council of Canada.

References

[1] J.M. Mendel, Uncertain Rule-based Fuzzy Logic Systems: Theory and Design, Prentice, Upper Saddle River, 2001.
[2] A. Celikyilmaz, I.B. Turksen, Type-2 fuzzy system models with improved fuzzy functions, in: Proceedings of 25th International Conference of the North American Fuzzy Information Processing Society-NAFIPS, 2007, pp. 140–145.
[3] O. Uncu, I.B. Turksen, Discrete interval-valued type 2 fuzzy system models using uncertainty in learning parameters, IEEE Transactions on Fuzzy Systems 15 (2007) 90–106.
[4] L.A. Zadeh, The concept of a linguistic variable and its application to approximate reasoning – 1, Information Sciences 8 (1975) 199–249.
[5] S. Coupland, R. John, New geometric inference techniques for type-2 fuzzy sets, International Journal of Approximate Reasoning 49 (1) (2008) 198–211.
[6] H. Hagras, Dynamical optimal training for interval type-2 fuzzy neural network, IEEE Transactions on Systems, Man and Cybernetics, B, Cybernetics 36 (5) (2006) 1206–1209.
[7] A. Celikyilmaz, I.B. Turksen, Fuzzy functions with support vector machines, Information Sciences 177 (2007) 5163–5177.
[8] A. Celikyilmaz, I.B. Turksen, Enhanced fuzzy system models with improved fuzzy clustering algorithm, IEEE Transactions on Fuzzy Systems 16 (3) (2008) 779–794.
[9] T. Takagi, M. Sugeno, Fuzzy identification of systems and its applications to modeling and control, IEEE Transactions on Systems, Man and Cybernetics SMC-15 (1) (1985) 116–132.
[10] I.B. Turksen, Fuzzy functions with LSE, Applied Soft Computing 8 (3) (2008) 1178–1188.
[11] C. Hwang, F.C.-H. Rhee, Uncertain fuzzy clustering: interval type-2 fuzzy approach to *c*-means, IEEE Transactions on Fuzzy Systems 15 (1) (2007) 107–120.
[12] N.K. Kasabov, Q. Song, DENFIS: dynamic evolving neural-fuzzy inference system and its application for time-series prediction, IEEE Transactions on Fuzzy Systems 10 (2) (2002) 144–154.
[13] F. Höppner, F. Klawonn, Improved fuzzy partitions for fuzzy regression models, International Journal of Approximate Reasoning 32 (2-3) (2003) 85–102.
[14] N.N. Karnik, J.M. Mendel, Q. Liang, Type-2 fuzzy logic systems, IEEE Transactions on Fuzzy Systems 7 (1999) 643–658.
[15] O. Cordon, F. Gomide, F. Herrera, F. Hoffmann, L. Magdalena, Ten years of genetic fuzzy systems: current framework and new trends, Fuzzy Sets and Systems 141 (2004) 5–31.
[16] J.R. Jang, Adaptive-network based fuzzy inference system, IEEE Transactions on System, Man, and Cybernetics, B, Cybernetics 3 (3) (1993) 665–685.
[17] R. Babuska, H.B. Verbruggen, Constructing fuzzy models by product space clustering, in: H. Hellendoorn, D. Driankow (Eds.), Fuzzy Model Identification: Selected Approaches, Springer, Berlin, Germany, 1997, pp. 53–90.
[18] W. Pedrycz, M. Reformat, Evolutionary fuzzy modeling, IEEE Transactions on Fuzzy Systems 11 (5) (2003) 652–665.
[19] J.C. Bezdek, Fuzzy Mathematics in Pattern Classification, Ph.D. Thesis, Applied Math. Center, Cornell University, Ithaca, 1973.
[20] S. Haykin, Neural Networks: A Comprehensive Foundation, second ed., Prentice Hall PTR, NJ, USA, 1998.
[21] I. Ozkan, I.B. Turksen, Upper and lower values for the level of fuzziness in FCM, Information Sciences 215 (2007) 5143–5152.
[22] A.J. Smola, B. Schölkopf, A tutorial on support vector regression, NeuroCOLT Technical Report NC-TR-98-030, Royal Holloway College, University of London, UK, 1998.
[23] J.T. Starczewski, Efficient triangular type-2 fuzzy logic systems, International Journal of Approximate Reasoning 50 (5) (2009) 799–811.
[24] C.L. Walker, E.A. Walker, Sets with type-2 operations, International Journal of Approximate Reasoning 50 (1) (2009) 63–71.
[25] M. Antonelli, P. Ducange, B. Lazzerini, F. Marcelloni, Learning concurrently partition granularities and rule bases of Mamdani fuzzy systems in a multi-objective evolutionary framework, International Journal of Approximate Reasoning 50 (7) (2009) 1066–1080.
[26] A. Fernández, M.J. del Jesus, F. Herrera, Hierarchical fuzzy rule based classification systems with genetic rule selection for imbalanced data-sets, International Journal of Approximate Reasoning 50 (3) (2009) 561–577.
[27] J.-N. Choi, S. Kwun Oh, W. Pedrycz, Structural and parametric design of fuzzy inference systems using hierarchical fair competition-based parallel genetic algorithms and information granulation, International Journal of Approximate Reasoning 49 (3) (2008) 631–648.
[28] P. Pulkkinen, H. Koivisto, Fuzzy classifier identification using decision tree and multiobjective evolutionary algorithms, International Journal of Approximate Reasoning 48 (2) (2008) 526–543.